International Journal of Computer Applications (0975 – 8887)
Volume 31– No.11, October 2011

# Advanced CAPTCHA technique using Hand Gesture based on SIFT

B.Srinivas
Assistant Professor
Computer Science and Engineering Department
MVGR College of Engineering
Vizianagaram, India

G.Kalyan Raju
Computer Science and Engineering Department
MVGR College of Engineering
Vizianagaram, India.

Dr. Koduganti Venkata Rao
School of Computing
Vignan University
Guntur

## ABSTRACT
CAPTCHA is widely used security technique employed to avoid automated form submissions or verify user as human. An alternative CAPTCHA method based on pattern matching and number recognition ability is proposed in this paper, which verifies user as human and prevents bots to intervene and spam applications or services. This method is based on user gestures which make it unique and secure. The biggest advantage of this new CAPTCHA technique is that it is simple and easy task conducted by user as it is language independent. It generates a random 4 character string number and shown to user. User should show gesture of particular characters in an order using computer webcam or using a mobile phone. A pattern matching algorithm is applied on those user images to identify gestures, and find matching. This method is very difficult to hack because designing a bot to identify gesture in image is not possible for now. Many experiments we conducted to prove accuracy of our technique.

## General Terms
Scale-Invariant feature transform, Completely Automated Public Turing test to tell Computers and Humans Apart, Pattern Recognition.

## Keywords
CAPTCHA, Hand Gesture, Security, SIFT, Key Points, reCAPTCHA.

## 1. INTRODUCTION
CAPTCHA (Completely Automated Tuning test to tell Computers and Humans Apart) is widely used security mechanism on the web to ensure that response is from a person [1]. An image of misrepresented letters is dynamically generated. The letters are part of image not a plain text CAPTCHA uses a type of challenge response test to determine that the response is not generated by the computer. Now a day's all sorts of websites are using small to large CAPTCHA's [1]. Commonly used CAPTCHA consists text based challenge is given to web clients to be typed in textbox where humans can pass the challenge but not the machine. For example a human can easily read misrepresented text shown in figure 1 but current programs cannot. CAPTCHA is mainly used to avoid spam. Spamming is performed on various public email provider sites and also on various forums and blogs. This comes into picture where users need to submit some content to a service or web applications

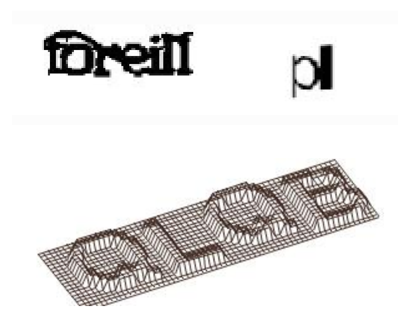


**Fig 1: General CAPTCHA images from Hotmail and rediff**.

Even though many types of CAPTCHA are implemented they fail at some time, for e.g., background CAPTCHA are easily hacked by using simple computer vision techniques [1]. CAPTCHA should be in such a way that it should be easily understandable by humans and should be easy to answer or interact with the process, it should be very hard for bots or machines to solve or it should be only understandable by humans, while bots cannot understand the process. And also there should be an easy process to generate and evaluate CAPTCHA and should not produce network overheads. Designing a CAPTCHA which satisfy these requirements is not so easy and that's the reason some times CAPTCHA [1] fail. Keeping in mind these systematic rules, our system performs 90% accurate process of identifying human. We generate simple clear images of characters that are not blended or twisted so it is really understandable by humans. Now we provide a new way of user interaction with CAPTCHA process, that is acquiring answer via image of user gesture either using webcam or mobile phone camera. Using this way bots cannot submit image, only user can submit gesture image [8, 9, 10.]. We perform a pattern matching algorithm and verify if user submitted image is a gesture of particular character [5, 6].Our system is highly intuitive and easily understandable by humans and user feel it more appealing than annoying traditional image based CAPTCHA [1]. We discuss user experience review of our system in Results and analysis section

## 2. RELATED WORK ON CAPTCHA

Many types of CAPTCHA are developed and implemented to prove particular subject is human or any kind of programmed bot. There have been various attempts at creating more accessible CAPTCHAs, including the use of JavaScript, mathematical questions ("how much is 1+1") and common sense questions ("what color is the sky on a clear day"). The most widely used CAPTCHAs are the so-called text-based schemes, which rely on sophisticated distortion of text images aimed at rendering them unrecognizable to the state of the art of pattern recognition programs[5]. The popularity of such schemes is due to the fact that they have many advantages, for example, being intuitive to users world-wide (the user task performed being just character recognition), having little localization issues (people in different countries all recognize Roman characters), and of good potential to provide strong security (e.g. the space a brute force attack has to search can be huge, if the scheme is properly designed).Internet spam is defined as "unsolicited commercial bulk e-mail", or junk mail, in other words advertisements that marketers blindly send to as many addresses as possible. It is widely accepted that the spam problem and the so called "Bots" have become a nuisance and must be defended against. Whereas individual anti-spam preventive measures and email address filtering may be used as a short term solution, there is a need for more comprehensive solutions such as HIPs and CAPTCHAs [1]. Alta Vista web site was among the first to use CAPTCHA to block the abusive automatic submission of URLs. Advanced efforts on HIPs have been made at the CMU. They have introduced the notion of CAPTCHA [1] and defined its mandatory properties. Several CAPTCHA systems (e.g. Gimpy, Bongo, Pix) are available to readers on their web site [1]. Over the past ten years, PARC and UC Berkeley have introduced new challenges. Mandatory Human Participation (MHP) is another kind of authentication scheme that uses a character-morphing algorithm to generate the character recognition puzzles.

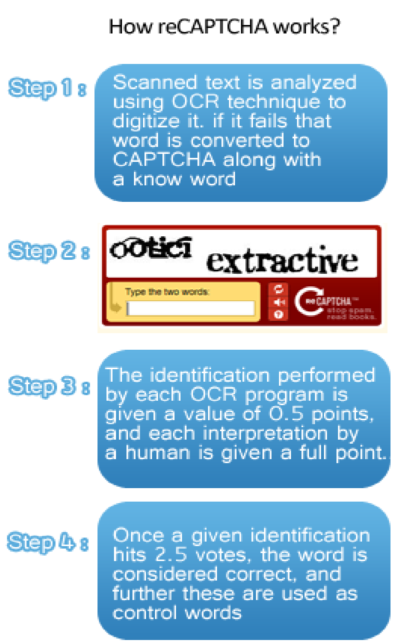



**Fig 2: Working of reCAPTCHA**

All the CAPTCHAs currently in commercial use take advantage of superior human ability in reading machine printed text[1]. Other algorithms use speech, facial features, and graphical Turing tests.

### 2.1. Disordered CAPTCHA

In this type of CAPTCHA particular string is processed in such a way that it is slightly or heavily distorted. Distortion may be angular or in proportion. Figure 2 shows a distorted CAPTCHA [1]. These types of CAPTCHA are easily understood by humans if they are slightly distorted. If it is heavily distorted it becomes difficult for users to identify string. Earlier day's yahoo used this kind of CAPTCHA called gimpy but this is prone to segmentation attacks so some techniques like adding a cross line with some angle and grouping of characters are implemented. Generally dictionary words are used as challenge with slight modifications.

### 2.2. Image based CAPTCHA

Image based CAPTCHA are of different types, some contains group of characters shown in foreground with background image containing noise. There are several types of noise commonly used with CAPTCHA [1] scripts to embarrass the recognition of symbols by spam-bots:

1. Pixel noise, sometimes of random colour, looks like an old grainy film or 3200 ISO images of your digital camera.
2. Lines, sometimes of random colour and angle, sometimes they form a kind of grid.
3. Rectangles and/or circles sometimes filled with colour.


**Fig 3: Image based CAPTCHA, Asirra, based on identifying cats and dogs in images**

Some other types of image based CAPTCHA include recognizing object in an image, recognizing shapes in image etc.

### 2.3. Limitations in existing system

However, some types of CAPTCHAs do not meet the criteria for a successful CAPTCHA [1]. A common type of CAPTCHA requires the user to type letters or digits from a distorted image that appears on the screen. Image recognition CAPTCHAs face many potential problems which have not been fully studied. It is difficult for a small site to acquire a large dictionary of images which an attacker does not have access to and without a means of automatically acquiring new labeled images, an image based challenge does not meet the definition of a CAPTCHA [1]. Segmentation attacks, dictionary attacks solve CAPTCHA easily by spam-bots. Converting image to grey scale and remove background noise, Reading session files to get CAPTCHA words and social engineering techniques like CAPTCHA solving business, annoying other site users by distributing and asking them to solve CAPTCHA[1]. These are some other techniques making present CAPTCHA system easily solvable by computers and getting ready to spam or take advantage of service.

## 3. PROPOSED SYSTEM

In our system we generate random character set and ask user to show gesture corresponding to certain character, User gesture is captured and processed to identify if it represents shown character, if gesture is correct CAPTCHA is solved and user is treated a human[1]. Figure 4 shows complete system setup. A random word of 4 character length is generated and characters are represented by C1, C2, C3, C4 user gesture images are captured as I1, I2, I3, I4 corresponding to C1,C2,C3,C4. We apply Robust Scale-invariant feature transform (SIFT) algorithm to find key-points of input images against database images. Now we find matching by calculating distance between database test image and input image and find Validity ratio or percentages of matching and find if two or more matches exist then we make slight variations to parameters like threshold and perform key-point calculation, matching and percentage calculation to find exact match and identify gesture.

### 3.1. Finding key-points using SIFT Algorithm:

Scale-invariant feature transform is an algorithm in computer vision to detect and describe local features in images. The algorithm was published by David Lowe in 1999

It is developed as a method to extract and describe key-points, which are robust to scale, rotation and change in illumination.

There are five steps to implement the SIFT algorithm:

1) Scale-space extrema detection: The first stage searches over scale space using a Difference of Gaussian (DoG) function to identify potential interest points that are invariant to scale and orientation.

2) Key-point localization: The location and the scale of each candidate point are determined and the keypoints are selected based on measures of stability.

3) Orientation assignment: One or more orientations are assigned to each key-point location based on local image gradient directions.

4) Key-point descriptor: A feature descriptor is created by computing the gradient magnitude and orientation at each image sample point in a region around the key-point location. These samples are then accumulated into orientation histograms summarizing the contents over 44 regions with 8 orientation bins. So each key-point has a 128-element feature.

5) The correspondence of feature points can be determined by taking the ratio of distance for the descriptor vector from the closest neighbor to the distance of the second closest.

Using this algorithm we read image and calculate key- points, descriptors and locations by applying threshold.

Descriptors given as P-by-128 matrix where p is number of key-points and each row gives an invariant descriptor for one of the p key-points. The descriptor is a vector of 128 values normalized to unit length.

Locations are P-by-4 matrix, in which each row has the 4 values for a key-point location (row, column, scale, orientation). The orientation is in the range [-PI, PI] radians.

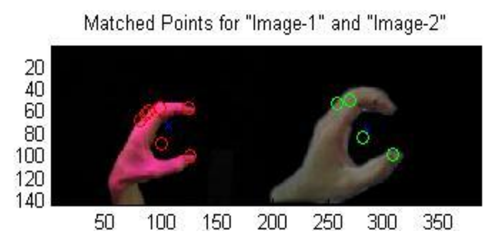



**Fig 5: SIFT Key-points Extraction, Image showing matched key-points between input image and database image.**

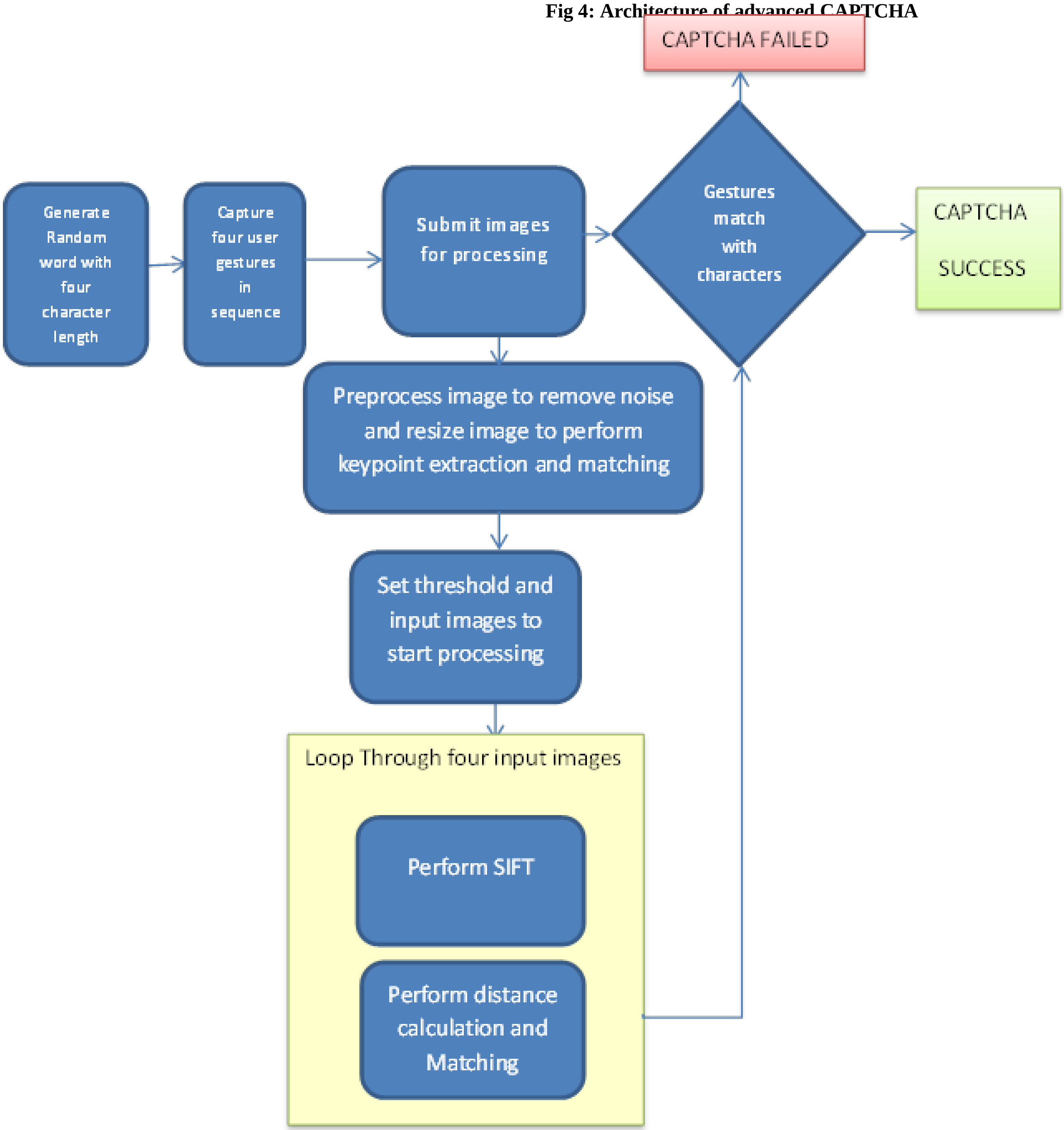
### 3.2. Finding images similarity using a distance calculation algorithm

After identifying key-points using SIFT, calculate the distances of the matched key-points to the center of key-points and construct a distance ratio array by summing distances of input image and test image. Now we verify if absolute of differences in distances sum of both images is below threshold value, and collect those points as they are valid matched key-points. If we have more than one match here we can decrease threshold and identify exactly a particular match [5]. We calculate validity ratio as by dividing number of valid key-points by number of matched key points

$$d_{T1} = \sum_{i=0}^{P} d_{I1} \quad - \ (1)$$

$$d_{T2} = \sum_{i=0}^{P} d_{I2} \quad - \ (2)$$

**Fig 4: Architecture of advanced CAPTCHA**




$$Ratios1 = [d_{11}/d_{T1} \;\; d_{21}/d_{t1} \;\; d_{31}/d_{t1}] \text{ -- (3)}$$

$$Ratios2 = [d_{12}/d_{T2} \;\; d_{22}/d_{t2} \;\; d_{32}/d_{t2}] \text{ -- (4)}$$

Distance = abs[Ratio1–Ratio2< MatchingThreshold] – (5)

ValidPoints = sum(distance) –(6)

The following gesture represents B, C, F, I, L, O respectively.

| Hand Gesture | Character Representation |
| --- | --- |
|  | INPUT B |

| | |
|---|---|
|  | INPUT C |
|  | INPUT F |
|  | INPUT I |
|  | INPUT L |
|  | INPUT O |

**Table I: The Real Time Hand gestures and Characters**

## 3.3 Hand Gesture processing

In this step we employee the SIFT Algorithm. SIFT finds the key-points from the input and compares with the gestures set in matched case proceeds further if not takes input repeatedly till gesture is satisfy.

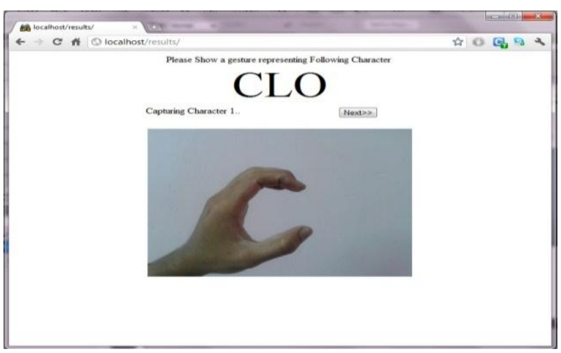



**Fig 6: Reading character1, C which is treated as input image1**


**Fig 7: Reading character 2, L which is treated as input image 2**

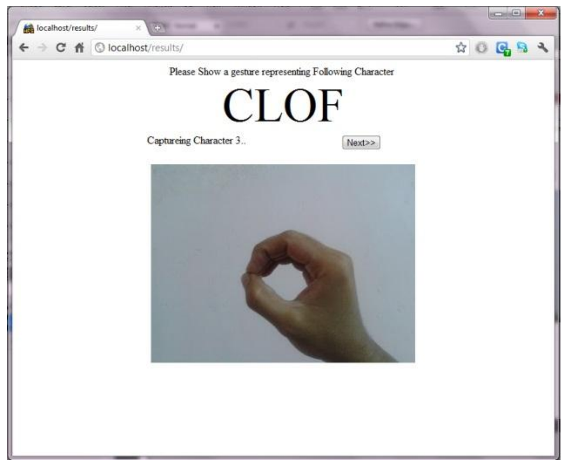



**Fig 8: Reading character 3, O which is treated as input image 3**


**Fig 9: Reading character 4, F which is treated as input image4**


**Fig 10: CAPTCHA system asking for user gesture input and completed processing**

## 4. RESULTS AND ANALYSIS

In order to prove the performance of our proposed system, we predefined the number of gestures from B, C, F, I, L, O, and create a hand gesture database. This database consists 200 training samples and 100 testing samples of 15 peoples.

We found 96% and above success rate when evaluating this system and able to find percentage of identifying a gesture that represent a character is 73%.

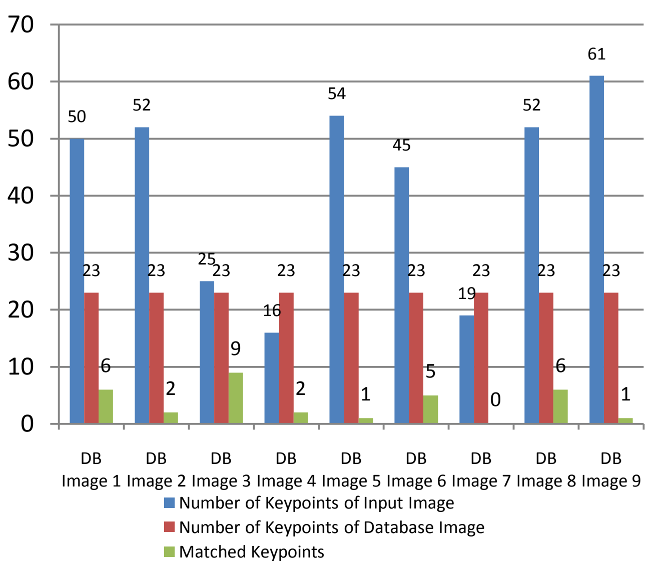



**Fig 11: comparison of key-points on given input with database for a single input image**

The Figure 5 depicts SIFT key-points analysis for a single input image we can easily identify DB Image 3 have more number of key-points matched with input image key-points

Figure 11. Graph showing Number of Key-points of Input Image, Database image and matched points for a single input image.

| Gesture Name | Testing Number | Success Number | Correct Rate |
|---|---|---|---|
| B | 150 | 148 | 98.7% |
| C | 150 | 147 | 98% |
| F | 150 | 145 | 96.7% |
| I | 150 | 148 | 98.7% |
| L | 150 | 147 | 98% |
| O | 150 | 147 | 98% |

**Table II. The results of classifier for the training set and testing set**

## 5. USER EXPERIENCE:

When presented this system to user, we got a huge response for its simplicity and intuitive ness; user is able to show gesture, correctly and is able to put 4 gestures correctly out of 6 gestures [6, 8, 10]. This system is quite easy for the user.

## 6. CONCLUSION AND FUTURE WORK

We proposed and implemented a Hand Gesture Based CAPTCHA system works efficiently. It is secure and very hard to crack, the experimental result shows that our system will work better input image has good resolution. This can be extended to a client server based model where CAPTCHA processing server is setup with an API to embed generated CAPTCHA on various websites and perform CAPTCHA human or bot verification.

## 7. REFERENCES

[1] L.v. Ahn, M. Blum, J. Langford, "Telling Humans and Computers Apart Automatically", Communications of the ACM available http://www.CAPTCHA.net/CAPTCHA_cacm.pdf. February 2004.

[2] M. Chew and H. Baird. Baffletext: A human interactive proof. Proc. SPIE-IST Electronic Imaging, Document Recognition and Retrieval, pages 305–316, January 2003.

[3] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. Proc. The Eighth IEEE International Conference on Computer Vision, 1:454–461, July 2001.

[4] M. Blum, L. von Ahn, J. Langford, and N. Hopper. The captcha project: Completely automatic public turing test to tell computers and humans apart. http://www.captcha.net ,

November 2000.

[5] M. Atallah, Y. Genin, and W. Szpankowski, "Pattern matching Image compression: Algorithmic and empirical results," IEEE Trans. Pattern Anal. Machine Intell., vol. 21, pp. 614–627, 1999.

[6] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4):366–379, 1997.

[7] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4):366–379, 1997.

[8] Y. Cui and J. J. Weng, "Hand Segmentation Using Learning-Based Prediction and Verification for Hand Sign Recognition," Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, Vt., pp. 88-93, Oct. 1996.

[9] T. Starner, J. Weaver and A. Pentland, "A wearable computer based American Sign Language recognizer," Proc. IEEE Symp. International Symposium on Wearable Computers (ISWC 97), IEEE Press, Oct.1997, pp. 130-137.

[10] B. Dorner, "Hand shape identification and tracking for sign language interpretation," IJCAI Workshop on Looking at People, 1993.

[11] H. Bunke and W. P. Handbook of Character Recognition and Document Image Analysis. World Scientific, 1997.

[12] Ritendra Datta, Jia Li, and James Wang. IMAGINATION:A Robust Image-based CAPTCHA Generation System.Proceedings of the 13th annual ACM internationalconference on Multimedia, 2005

[13] Greg Mori and Jitendra Malik. Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In Computer Vission and Pattern Recognition, 2003.

[14] Andy Schlaikjer, A Dual-Use Speech CAPTCHA: Aiding Visually Impaired Web Users While Providing Transcriptions and Audio Streams. CMU-LTI-07-014. http://www.lti.cs.cmu.edu/

[15] ReCAPTCHA:Stop Spam Read Books (2007), Retrieved on May 7, 2011 at http://recaptcha.net/

[16] BBC news PC stripper helps spam to spread. http:// news.bbc.co.uk/2/hi/technology/7067962.stm.

[17] Ticketmaster, LLC v. RMG Technologies, Inc., et al 507 F.Supp.2d 1096 (C.D. Ca., October 16, 2007).

[18] E. Bursztein, S. Bethard, J. C. Mitchell, D. Jurafsky, and C. Fabry. How good are humans at solving CAPTCHAs? a large scale evaluation. In IEEE S&P '10, 2010.

[19] M. Chew and D. Tygar. Image recognition CAPTCHAs. In Information Security, 7th International Conference, ISC 2004, pages 268–279. Springer, 2004.

[20] J. Elson, J. R. Douceur, J. Howell, and J. Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In CCS '07, pages 366–374, New York, NY, USA, 2007. ACM.

[21] C. Fleizach, M. Liljenstam, P. Johansson, G. M. Voelker, and A. M´ehes. Can You Infect Me Now? Malware Propagation in Mobile Phone Networks. In Proceedings of the ACM Workshop on Recurring Malcode (WORM), Washington D.C., Nov. 2007.

[22] A. Hindle, M. W. Godfrey, and R. C. Holt. Reverse Engineering CAPTCHAs. In Proc. of the 15th Working Conference on Reverse Engineering, pages 59–68, 2008.

[23] L. Jassin-O'Rourke Group. Global Apparel Manufacturing Labor Cost Analysis 2008. http://www.tammonline.com/files/ GlobalApparelLaborCostSummary2008.pdf, 2008.

[24] R. F. Jonell Baltazar, Joey Costoya. The heart of KOOBFACE: C&C and social network propagation. http://us.trendmicro.com/us/trendwatch/research-and-analysis/white-papers-and-articles/, October 2009.

## 8. AUTHORS PROFILE

**Srinivas Baggam** received the M.Tech (Computer Science & Engineering) from R.V.R & J.C college of Engineering, Guntur, Affiliated to Acharya Nagarjuna University. Currently working as an Assistant Professor in M.V.G.R. College of Engineering. He got two and half years of Industrial and Three years in teaching experience.

**Kalyan Raju Gutti** completed B.Tech (Electronics and Communication Engineering) from M.V.G.R College of engineering, Vizianagaram, Affiliated to JNTU Kakinada.

**Dr. Koduganti Venkata Rao** received Ph.D in Computer Science and Engineering from Andhra University, M.Tech in (Computer Science and Technology) from Andhra University and M.Sc (computer science) from Nagarjuna University, 2008, 1999, 1994 respectively. Currently working as a Professor School of Computing,Vignan University, Guntur.